

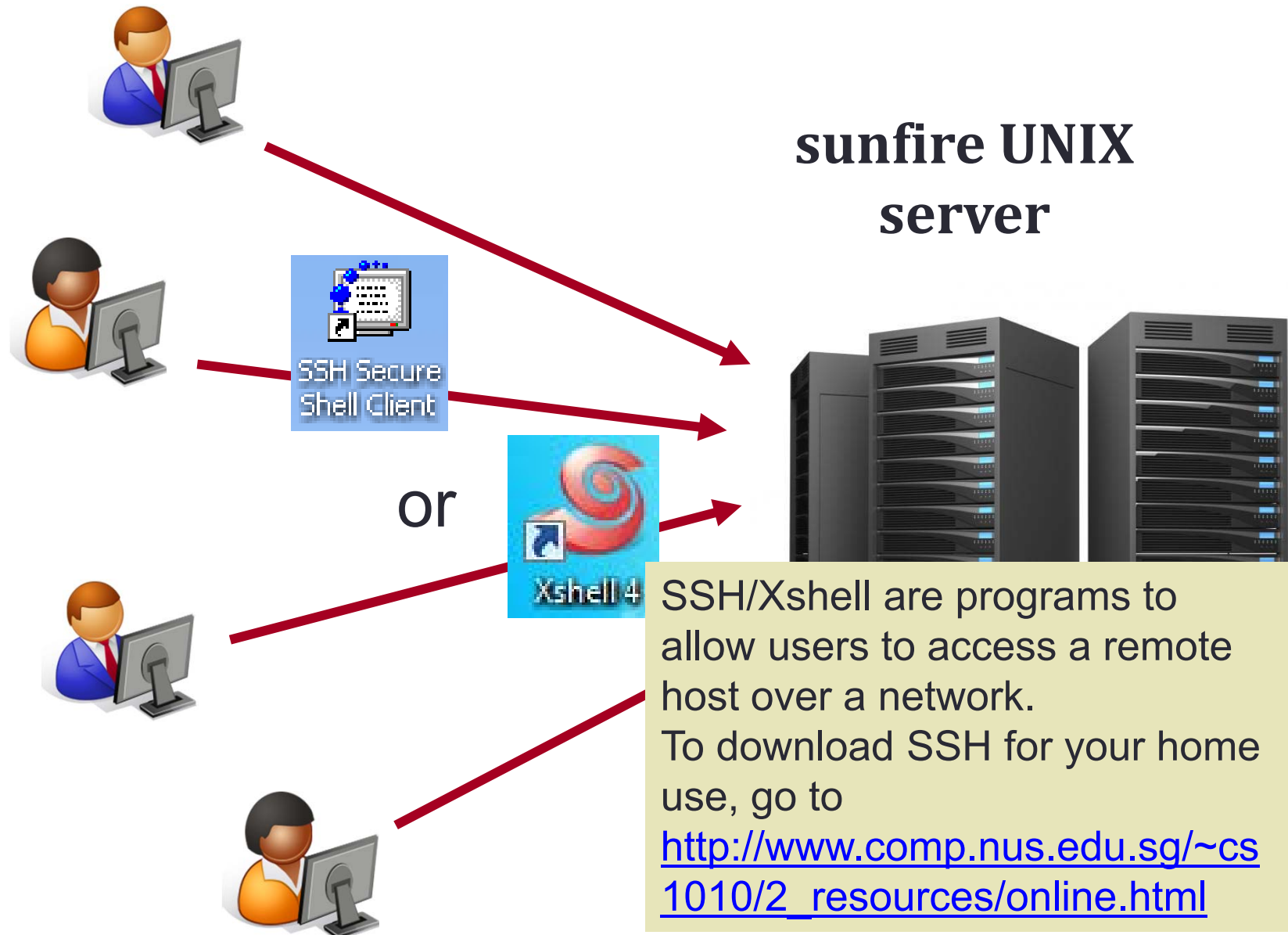
CS1010 Programming Environment (1/2)

- UNIX system – the sunfire server
- Every SoC student or student taking an SoC programming module can apply for a UNIX account
- To login to sunfire server, you need your SoC UNIX account user-name and password.
- If you don't have a UNIX account yet, go to this link to create one (same link if you have forgotten your UNIX password):

<https://mysoc.nus.edu.sg/~newacct>

CS1010 Programming Environment (2/2)

- You can do many things with your sunfire account:
 - Eg: Your account comes with paper quota
 - see <https://docs.comp.nus.edu.sg/node/1732> for your print quota allocation
 - Some treat their sunfire account as a backup harddisk
 - Refer to SoC Computing Facilities web page for more general information
<https://docs.comp.nus.edu.sg/cf/>



Logging into sunfire (1/2)

1. Look for the **SSH Secure Shell Client** icon or **Xshell** icon on your desktop, and double click on it. We shall assume you are using the former here.



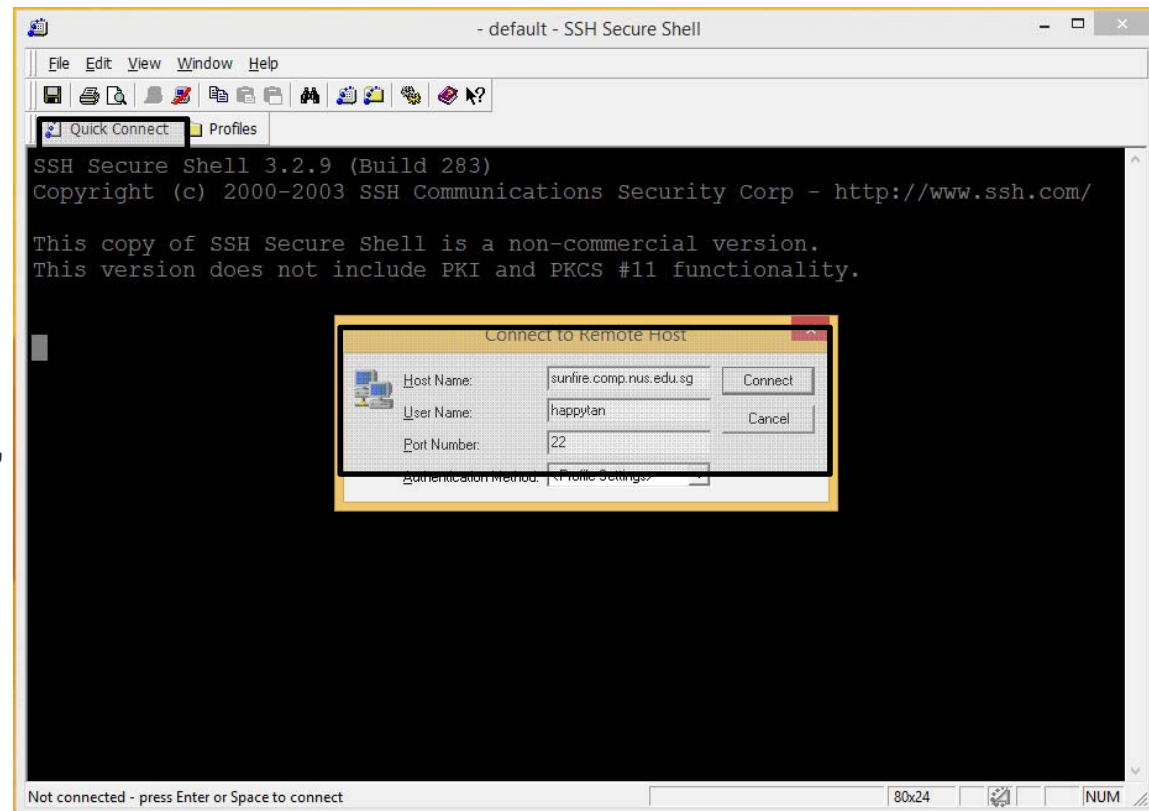
or



2. Click on “**Quick Connect**” to get the pop-up window.

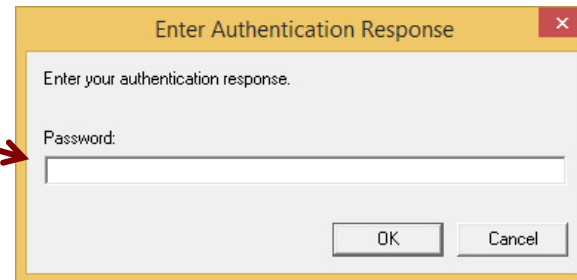
Enter “**sunfire**” for Host Name if connecting within campus or
“**sunfire.comp.nus.edu.sg**” if connecting from off campus

Enter your **UNIX id** as User Name.



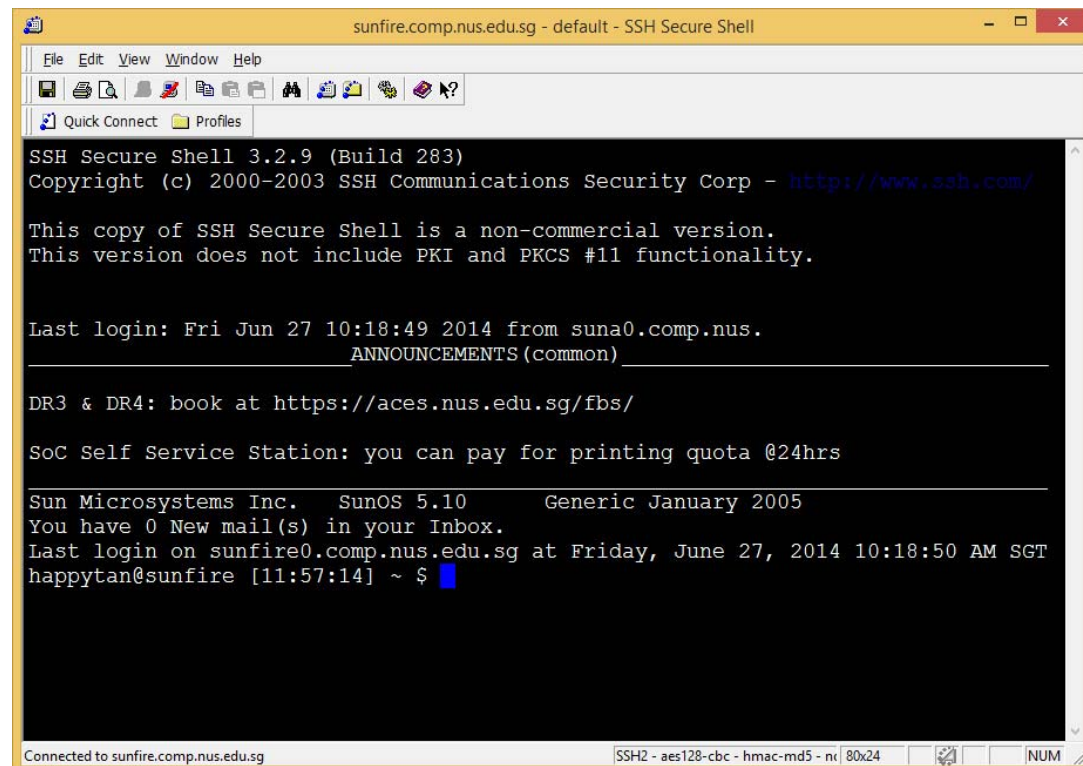
Logging into sunfire (2/2)

3. Enter your UNIX password.



4. Once you log in successfully, you will see this screen (actual display may vary).

5. To log out from your UNIX account, type “**exit**” or “**logout**”.



Trying out some UNIX commands

- Type 'ls' (list) to list out the files in your directory (**directory = folders in Windows**)
- You see no list because your account is brand new. There are no files in there.
- Type 'pwd' (print working directory) to show the pathname of your current directory
 - An example output: `/root/home/h/happytan`

ls and **pwd** are just two UNIX commands.
UNIX commands are case-sensitive.

Pathname

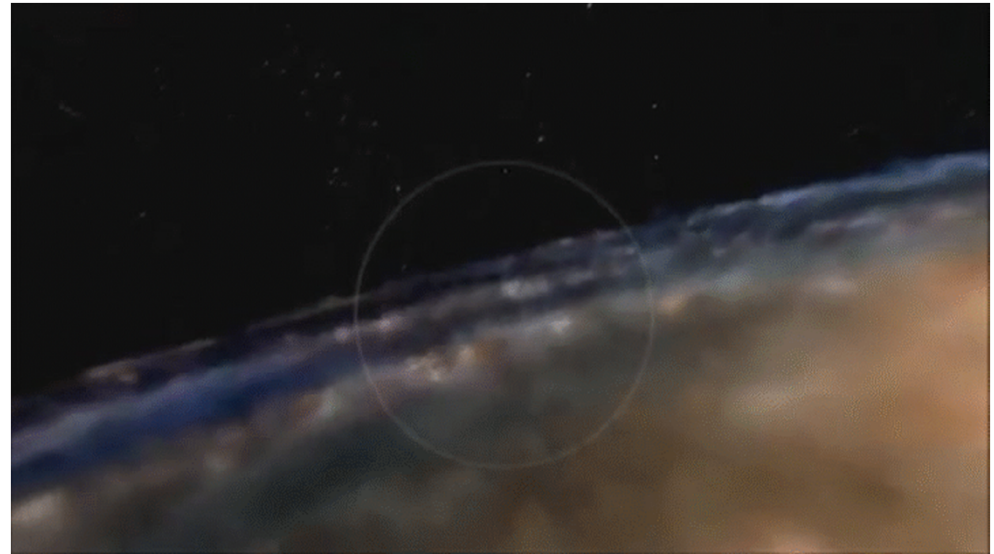
- It's like how to describe where you are now in

- The Universe
- Earth
- Singapore
- NUS
- Then your path is:

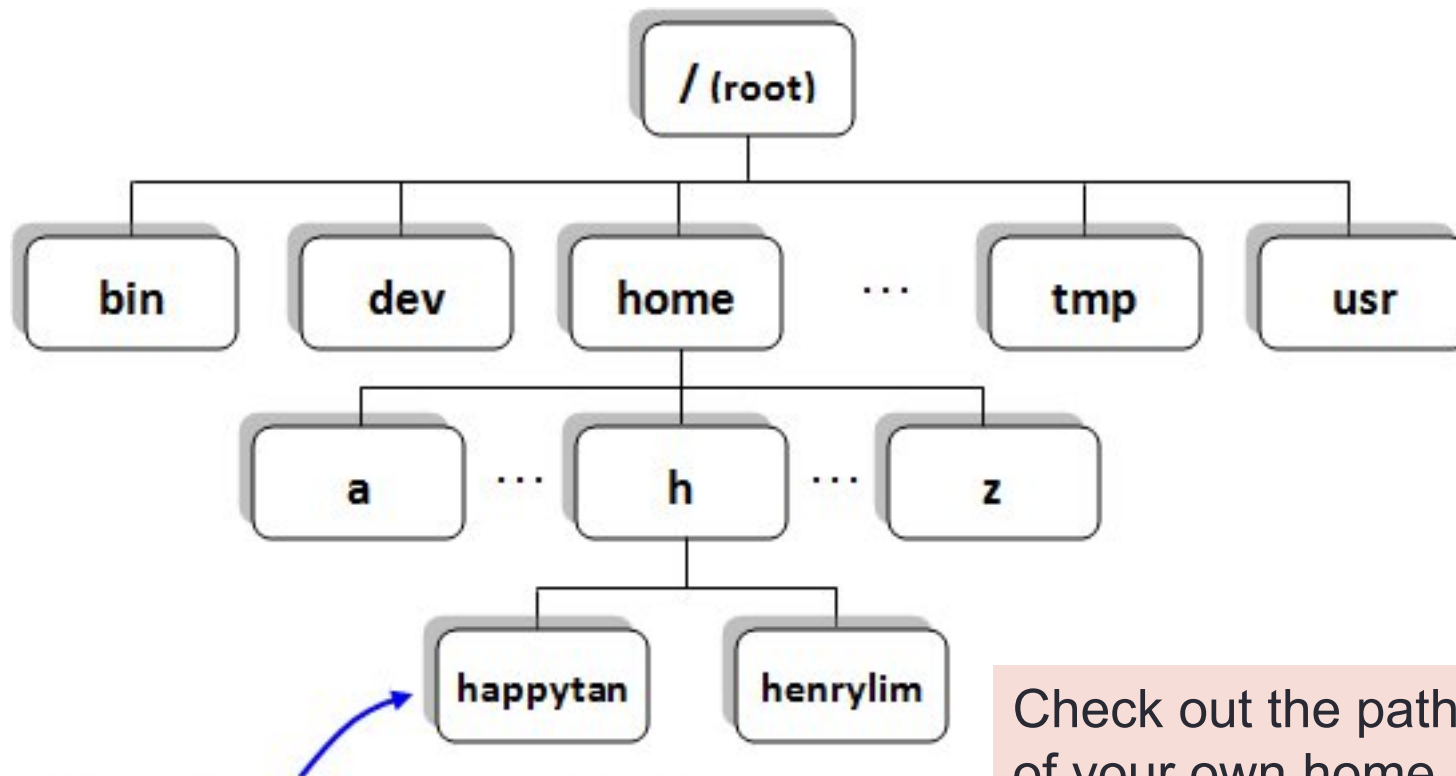
`/Universe/Earth/Singapore/NUS`

- So as in the computer:

`/root/home/h/happytan`



File Directories in sunfire (1/2)



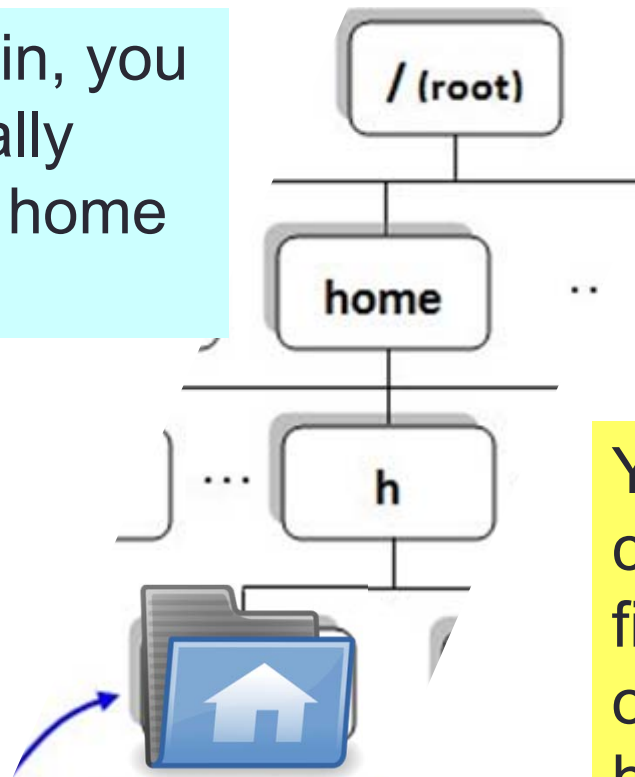
Home directory of user happytan.

/home/h/happytan

Check out the pathname of your own home directory by typing **'pwd'**

File Directories in sunfire (2/2)

When you log in, you are automatically placed in your home directory.



You are allowed to create/modify/remove files or subdirectories only under your home directory.

`/root/home/h/happytan`

Setting up your UNIX account



- As your new account is currently bare, run the following set-up to configure your account:

1. `~cs1010/workshop/setup`

(enter **y** when prompted)

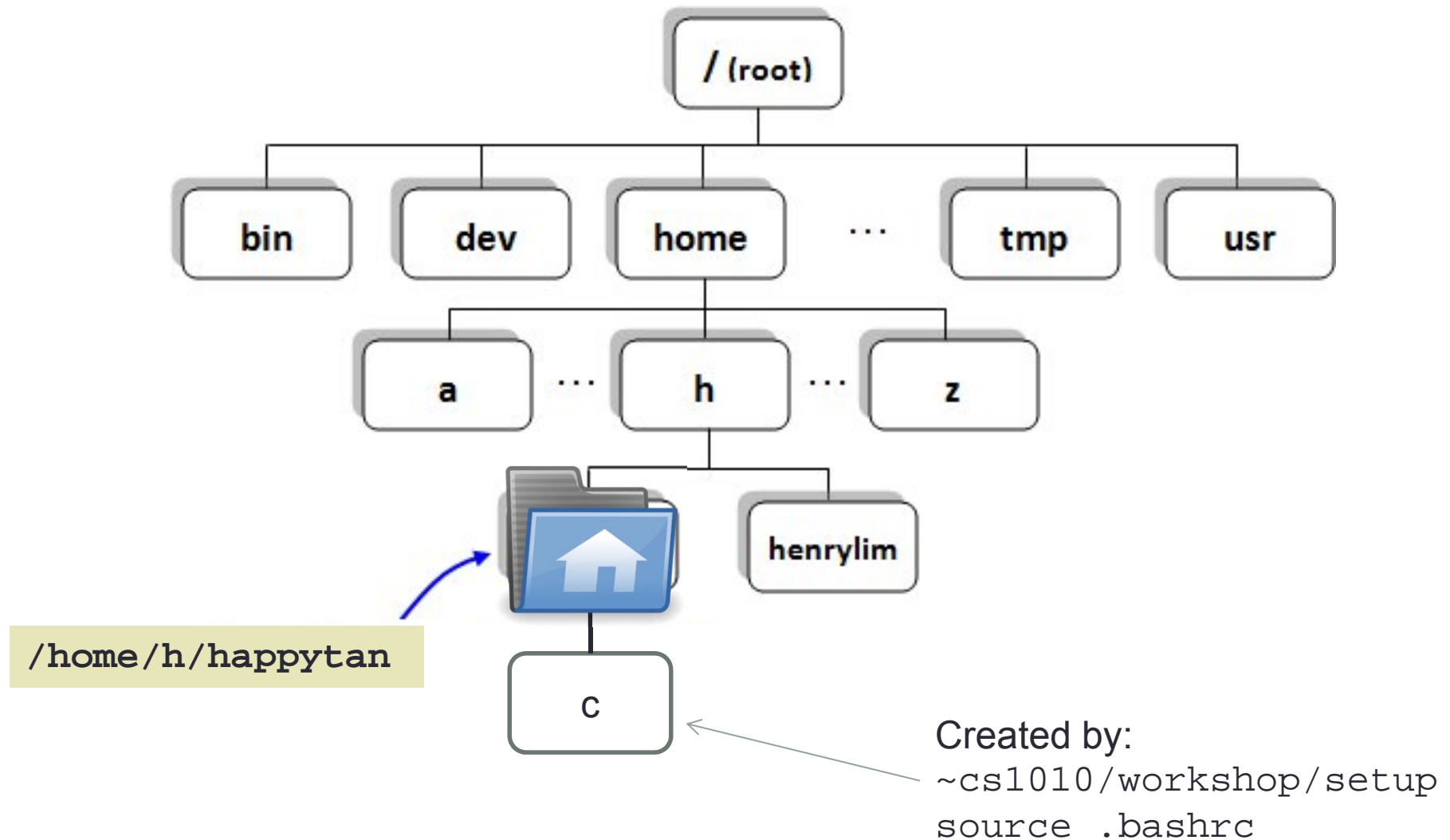
2. `source .bashrc`

(no response from the system is good news!)

You need to
do this only
ONCE.

- (1) does the following in your home directory
 - Creates a 'c' subdirectory and puts a few C programs into the 'c' subdirectory
 - Copies a number of system files into the home directory, including **.vimrc** (vim configuration file)

Basic UNIX Commands (4/3)



Basic UNIX Commands (1/4)

- In UNIX, typically you do a lot of typing but much less mouse clicking, compared with other operating systems like Windows
- UNIX commands are case sensitive
- Practice is the best way to recognize UNIX commands. Gradually you will be more and more familiar with UNIX commands – so don't worry too much at the beginning
- In sunfire, you can use the up ↑ and down ↓ arrows to select (and optionally modify) a previous command in the command log

Basic UNIX Commands (2/4)

- Following the “Getting Started with UNIX and CodeCrunch” document

(http://www.comp.nus.edu.sg/~cs1010/labs/2016s1/intro_lab/gettingStarted.html), your lecturer will go through these basic UNIX commands with you in class. (We will introduce CodeCrunch in the next lesson.)

Directory command	Description
pwd	Print W orking D irectory
ls	L i S t files in current directory
cd	C hange D irectory
mkdir	M a K e a sub D i R ectory
rmdir	R e M ove an empty sub D i R ectory

File command	Description
cp	C o P y file
mv	M o V e file, also to rename file
rm	R e M ove file
cat	C A T enate file (to view a file)

Basic UNIX Commands (3/4)

■ Command **options**

- Many UNIX commands come with options, preceded by '-'

```
happytan@sunfire [] ~ $ ls  
c
```

The plain **ls** command

```
happytan@sunfire [] ~ $ ls -F  
/c
```

-F prefixes directory name with /

-l displays info in long format

```
happytan@sunfire [] ~ $ ls -l  
drwx----- 2 happytan soc06 4096 Jun 27 12:58 c
```

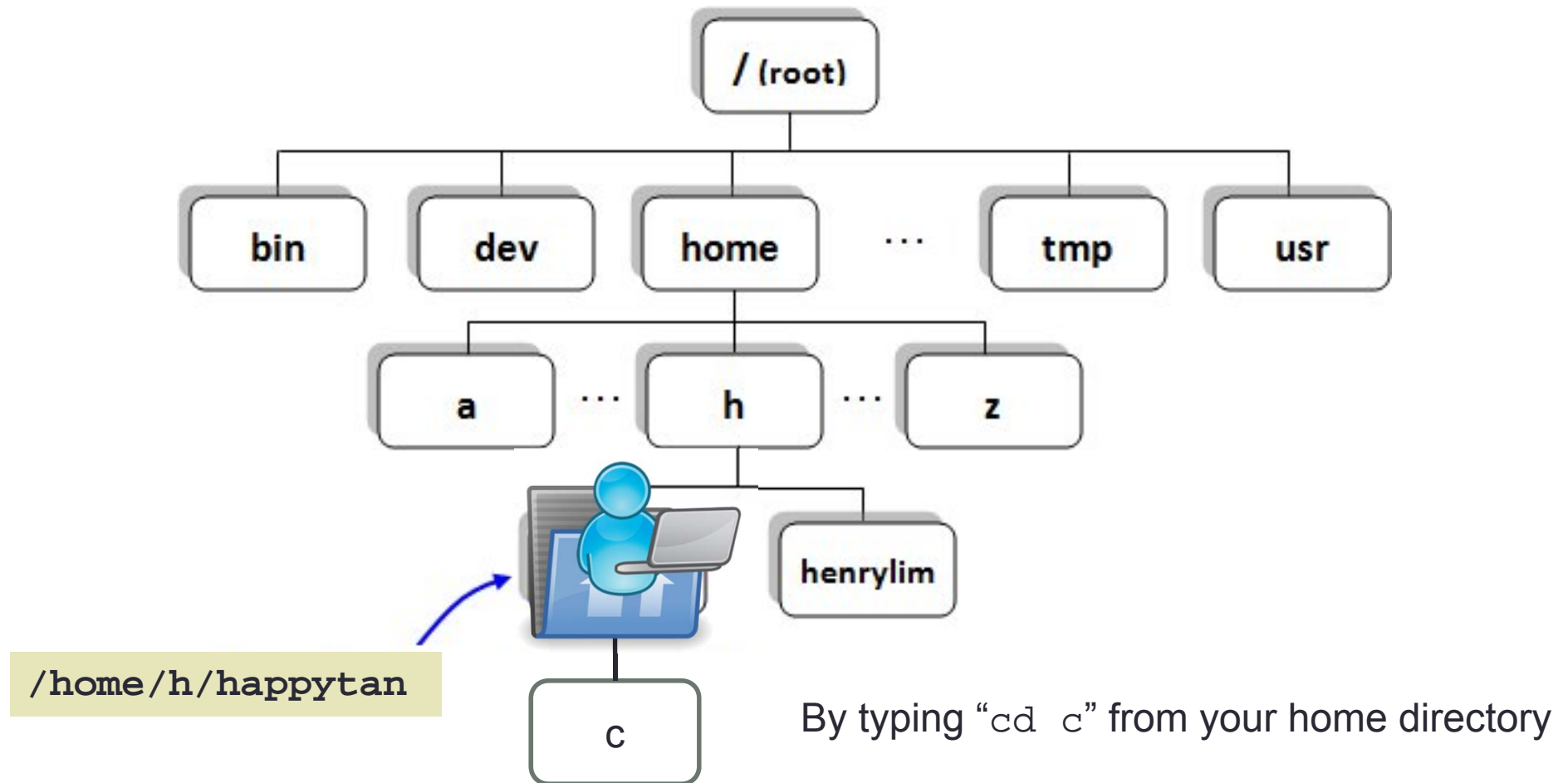
```
happytan@sunfire [] ~ $ ls -a  
.  ..  .bashrc  .vimrc  c
```

-a displays hidden files (files beginning with '.' in their names)

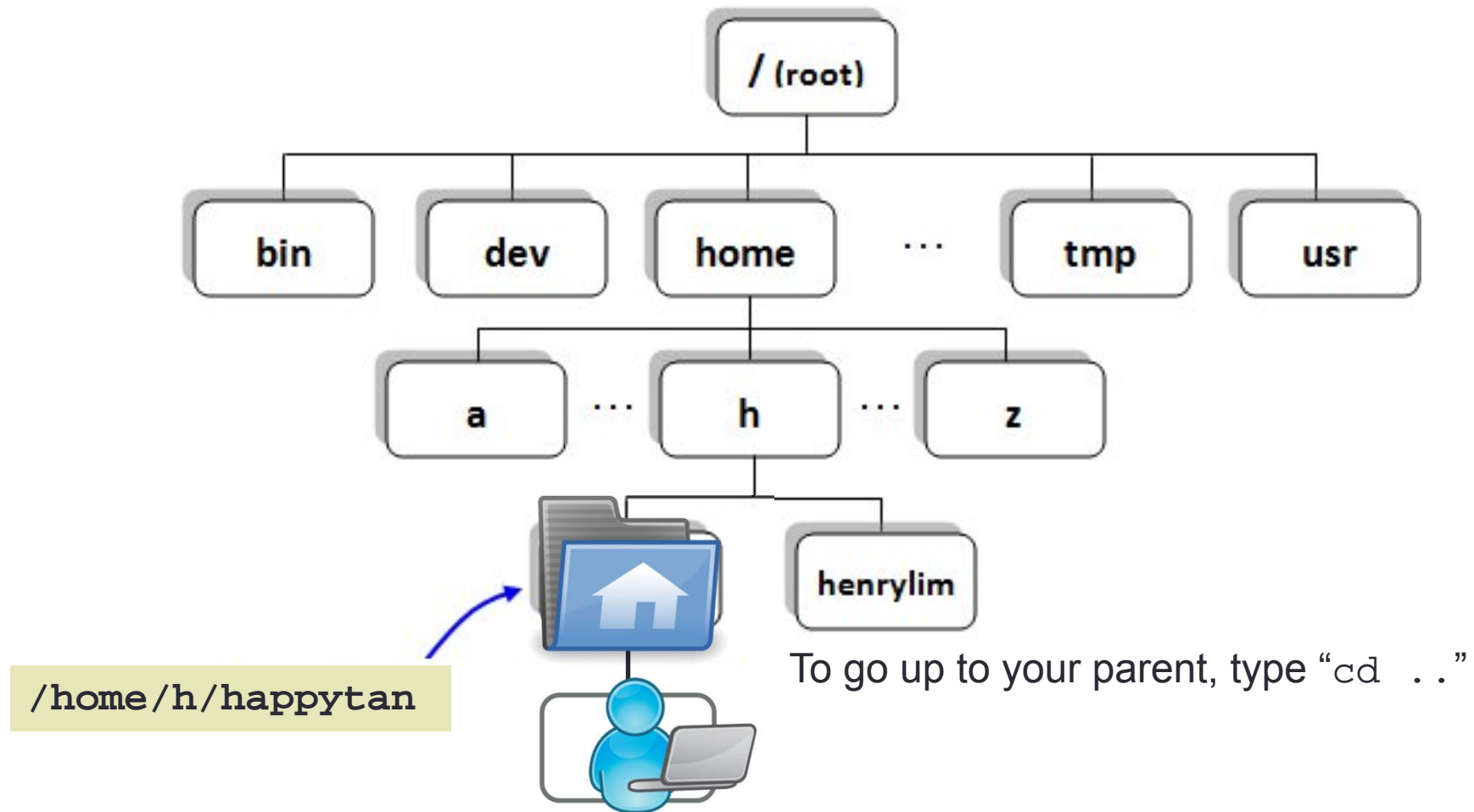
```
happytan@sunfire [] ~ $ ls -al  
drwx----- 3 happytan soc06 4096 Jun 30 08:45 .  
drwxr-xr-x 215 happytan root 8192 Jun 13 12:58 ..  
-rwx----- 1 happytan soc06 434 Jun 27 12:45 .bashrc  
-rwx----- 1 happytan soc06 237 Jun 27 12:45 .vimrc  
drwx----- 2 happytan soc06 4096 Jun 27 12:58 c
```

Options may be combined:
ls -al or **ls -a -l**

Basic UNIX Commands (4/3)



Basic UNIX Commands (4/3)



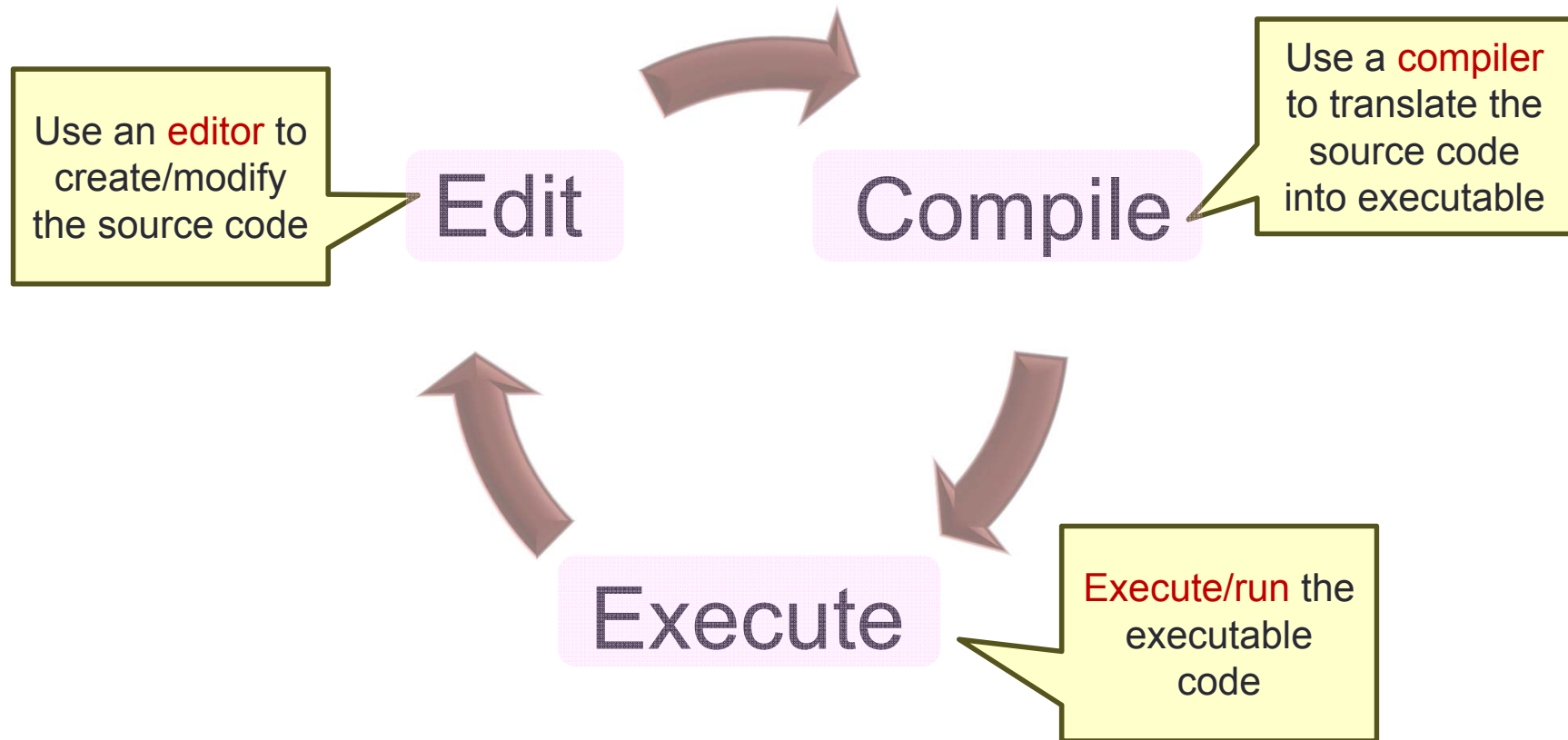
Basic UNIX Commands (4/4)

- Help using **man** command ('man' stands for 'manual')
 - Type **man command** to find out more about a certain command
 - Eg: man ls
 - Press <spacebar> to read next screen, or enter 'q' to quit.
- Filename auto-filling
 - Provides auto-filling of filenames, handy for very long filenames
 - Press <tab> for system to fill out the rest of the filename (as much as it can)

```
happytan@sunfire [] ~ $ cd c
happytan@sunfire [] ~/c $ ls
example1.c  example2.c  example3.c
happytan@sunfire [] ~/c $ cat e ← press <spacebar> after typing 'e' and observe
```

System managed to fill filename up to 'example' and stopped, because there are 3 filenames that begin with 'example'. Type '1', '2', or '3' and press <tab> for system to fill the whole filename, then press <enter>.

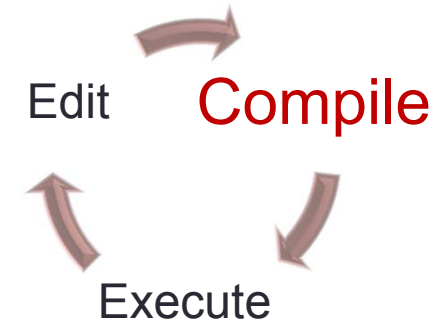
The Edit, Compile and Execute Cycle



Process is iterative

Compiling C programs (1/3)

- We use the C compiler **gcc** in sunfire



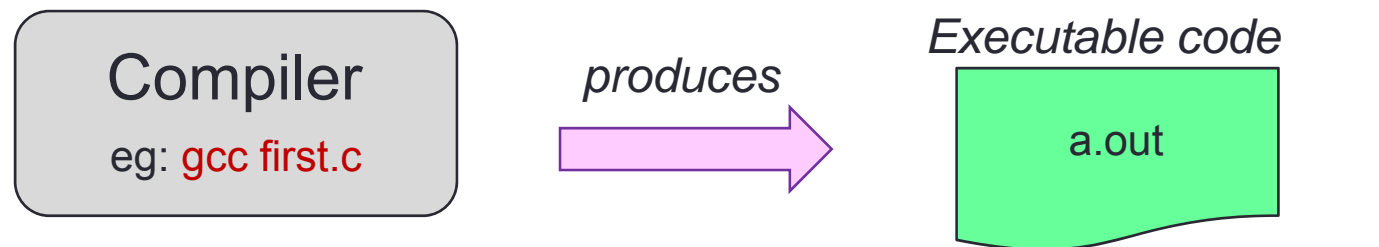
- Let's try compile the file "example2.c"

```
happytan@sunfire [] ~/c $ gcc example2.c
```

- If nothing comes out... good....
- See what is new in your directory by "ls"
- Try running that file by typing its name
- Next you can try to compile "example2.c"
 - Oops....

Compiling C programs (1/3)

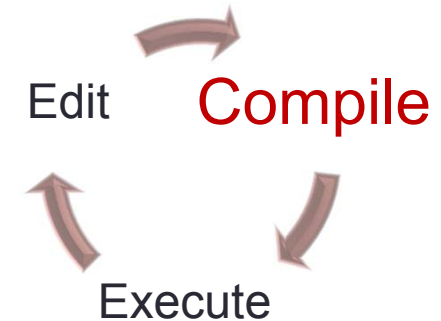
- We use the C compiler **gcc** in sunfire



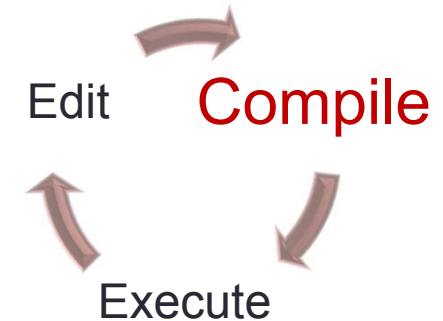
- Advisable to add the option **-Wall** (warnings all) for beginners:
gcc -Wall first.c
- If there are compilation errors/warnings, you need to edit the source code **first.c** again (**vim first.c**), and re-compile (**gcc -Wall first.c**), until your code is clear of compilation errors/warnings.
- Remember to add option **-lm** if your C program uses math functions
 - Example: **gcc -Wall -lm example1.c**
- Type **ls** to check that you have the executable code **a.out**

Compiling C programs (2/3)

- The executable file has the default name **a.out**. However, all filenames in a directory must be unique, hence there can only be one **a.out** in a directory.
- Since you have many C source codes in a directory (eg: example1.c, example2.c, example3.c), you might want to have their corresponding executable files all in the same directory, appropriately named.
- Two approaches:
 1. Rename **a.out** after compilation
 2. Indicate the desired name of the executable file during compilation



Compiling C programs (3/3)



1. Rename **a.out** after compilation

```
happytan@sunfire [] ~/c $ gcc -Wall -lm example1.c
happytan@sunfire [] ~/c $ mv a.out example1
happytan@sunfire [] ~/c $ gcc -Wall example2.c
happytan@sunfire [] ~/c $ mv a.out example2
happytan@sunfire [] ~/c $ gcc -Wall example3.c
happytan@sunfire [] ~/c $ mv a.out example3
```

Executable files are named example1, example2, example3.

2. Indicate the desired name of the executable file during compilation using the **'-o'** option

```
happytan@sunfire [] ~/c $ gcc -Wall -lm example1.c -o example1
happytan@sunfire [] ~/c $ gcc -Wall example2.c -o example2
happytan@sunfire [] ~/c $ gcc -Wall example3.c -o example3
```



Be careful not to overwrite the source code accidentally!
The following will replace the source code with the executable file, which is called example1.c now! The source code cannot be recovered!

WRONG WAY

```
happytan@sunfire [] ~/c $ gcc -Wall -lm example1.c -o example1.c
```

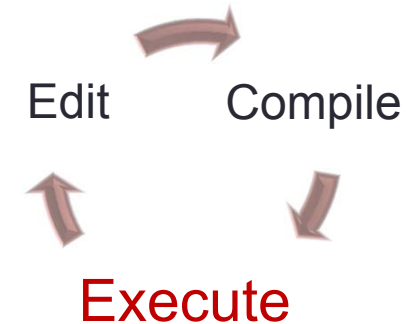
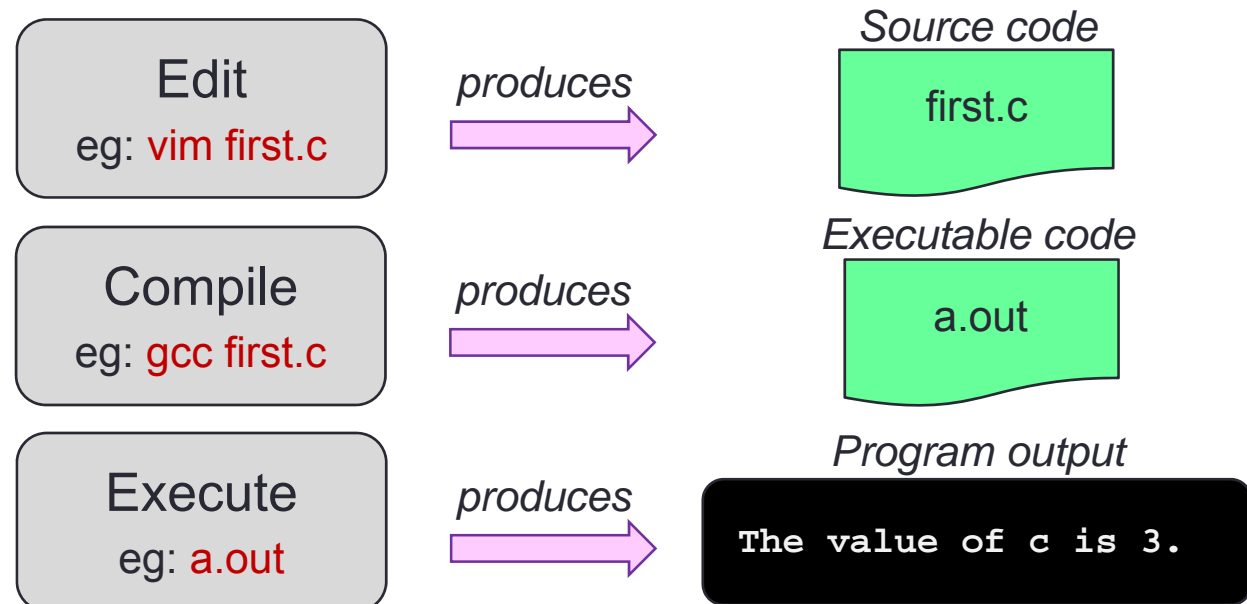
Executing C programs

- Executing a C program is simple – just type the name of the executable file

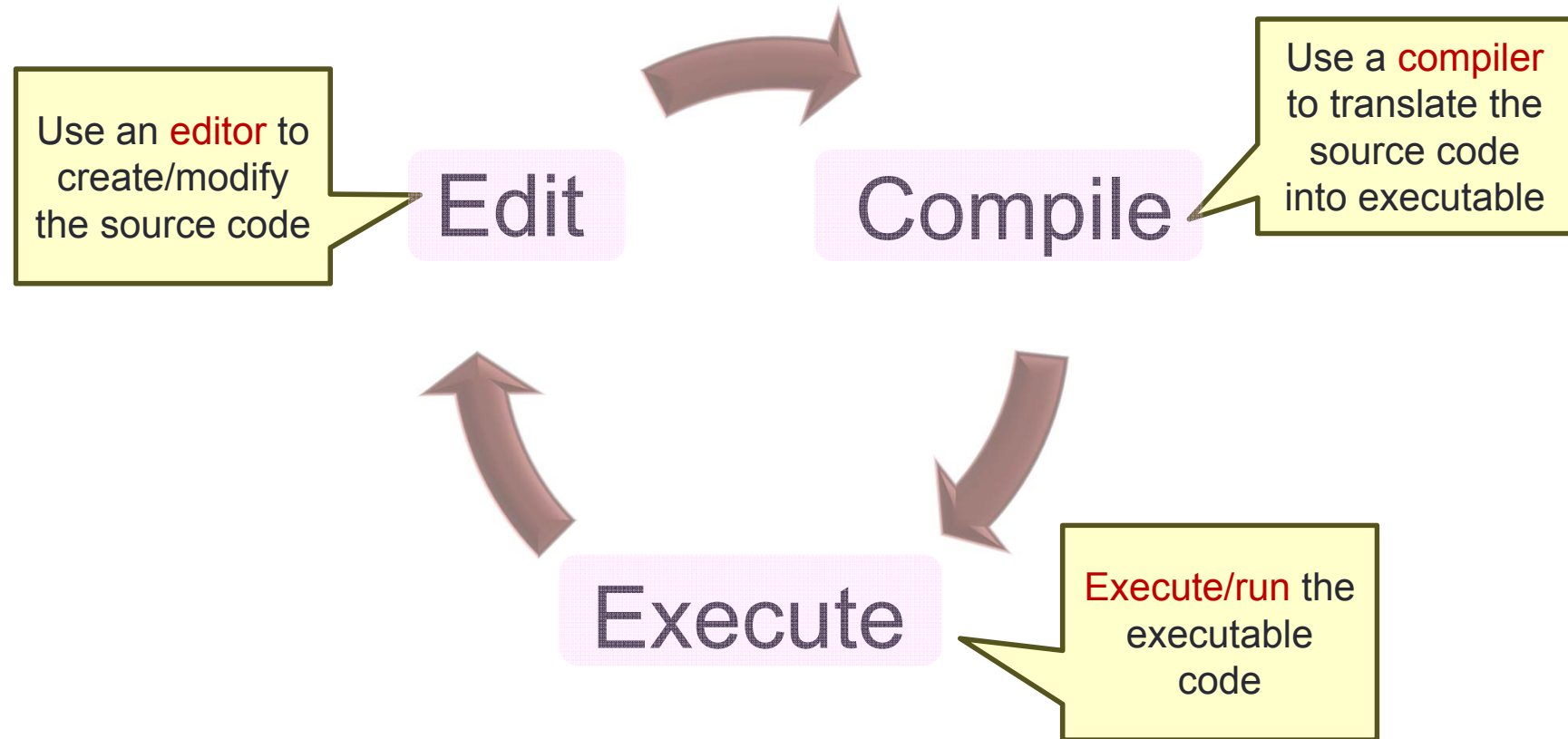
To run the executable file **example1**:

```
happytan@sunfire [] ~/c $ example1  
The distance between the 2 points is 3.61
```

- The **Edit – Compile – Execute** process



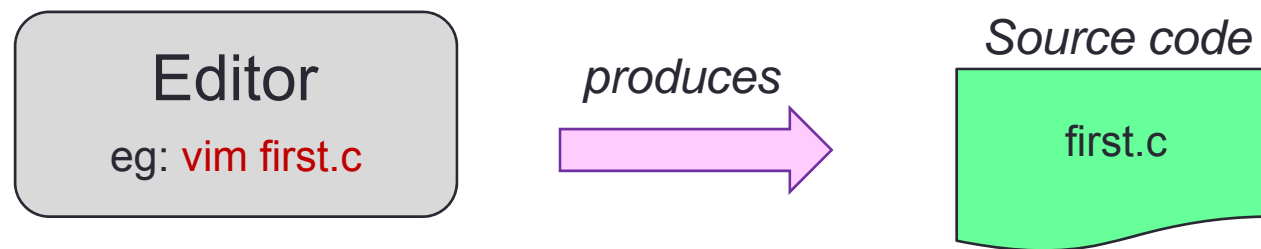
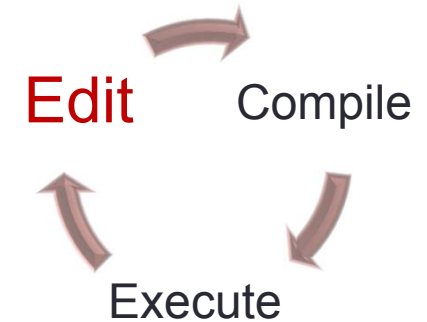
The Edit, Compile and Execute Cycle



Process is iterative

Editing C source codes (1/3)

- We use a text editor to create/modify C programs (source codes)
- We will use the **vim** editor



- **vim** is a powerful text editor. It has many modes
 - **Command mode**: for issuing vim commands
 - **Insert mode**: for typing in text
- To switch between command mode and insert mode
 - Type **i** in command mode to get into insert mode
 - Press **<esc>** key in whatever mode to get into command mode

Vim is like....



- Chopper!!
 - With different modes!

Editing C source codes (2/3)

- Use vim to create this C program `first.c`

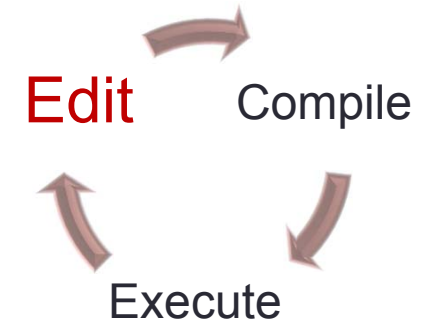
```
#include <stdio.h>

int main(void) {
    printf(" @..@\n");

    return 0;
}
```

- To start, type

```
vim first.c
```



Super Simple Basic vim Commands

- `<esc>` to return to command **mode**
- `"i"` to go into insert **mode**, type `<esc>` to get out
- `"x"` for delete
- `"h"`, `"j"`, `"k"`, or `"l"` for moving left, up, down, right
- `":"` to enter line command **mode**, after that
 - `"w"` for writing a file
 - `"q"` for quitting vim
 - or instead, you can type `"wq"`
- Or altogether, you just type `":wq"`

More Advanced vim Commands

- “yy” copy a whole line
 - “dd” cut a whole line
- “p” paste
- “a” to append mode, equivalent to inserting after the current position
- If you are in the command mode, you can type a number n and then the command. Then the command will repeat n times.
 - E.g. if you type “dd”, “5”, “p” this will paste the line 5 times

Editing C source codes (2/3)

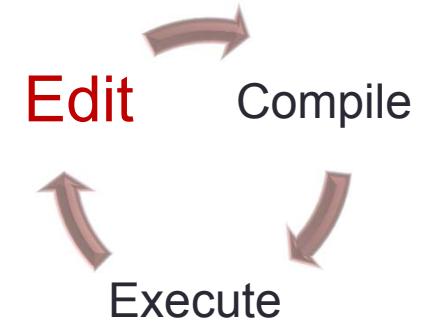
- Use vim to create this C program **first.c**

```
#include <stdio.h>

int main(void) {
    printf(" @..@\n");

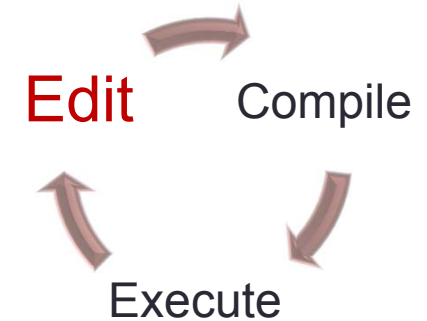
    return 0;
}
```

- Then try to compile and execute



Editing C source codes (2/3)

- Now, try using vim to create this C program **first.c**



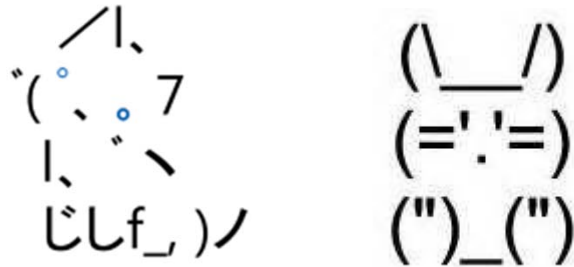
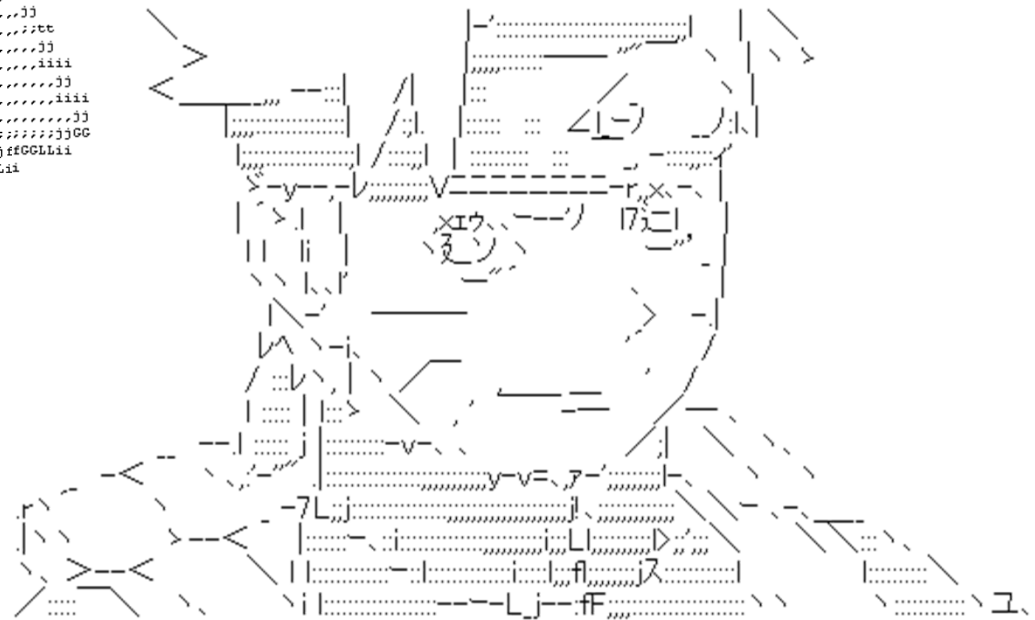
```
#include <stdio.h>

int main(void) {
    printf("  @..@\n");
    printf(" (----)\n");
    printf("( >__< )\n");
    printf("^^  ~~  ^^ \n");
    return 0;
}
```

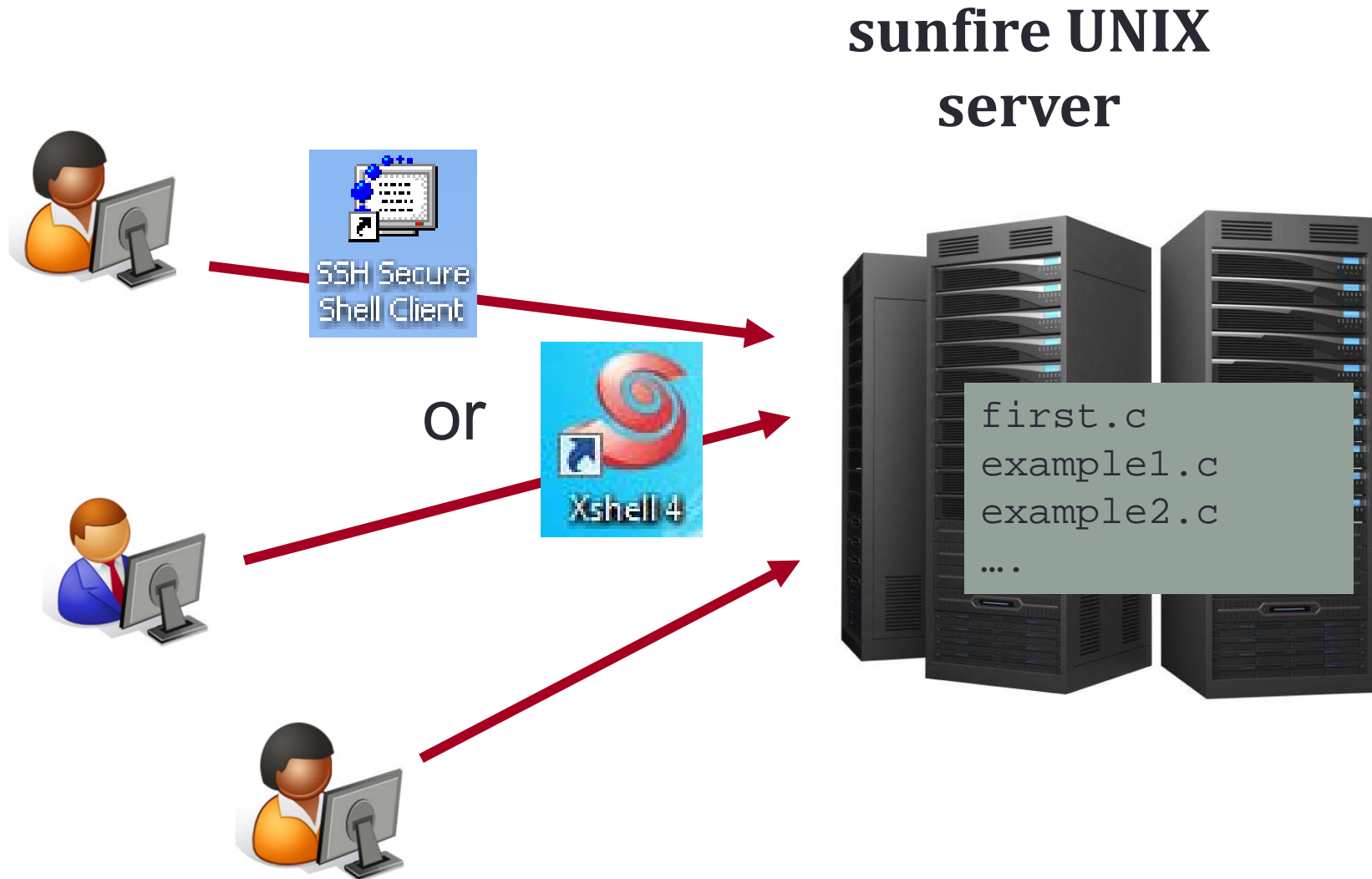
```
  @..@
 (----)
( >__< )
 ^^  ~~  ^^
```

ASCII art:

ASCII Art

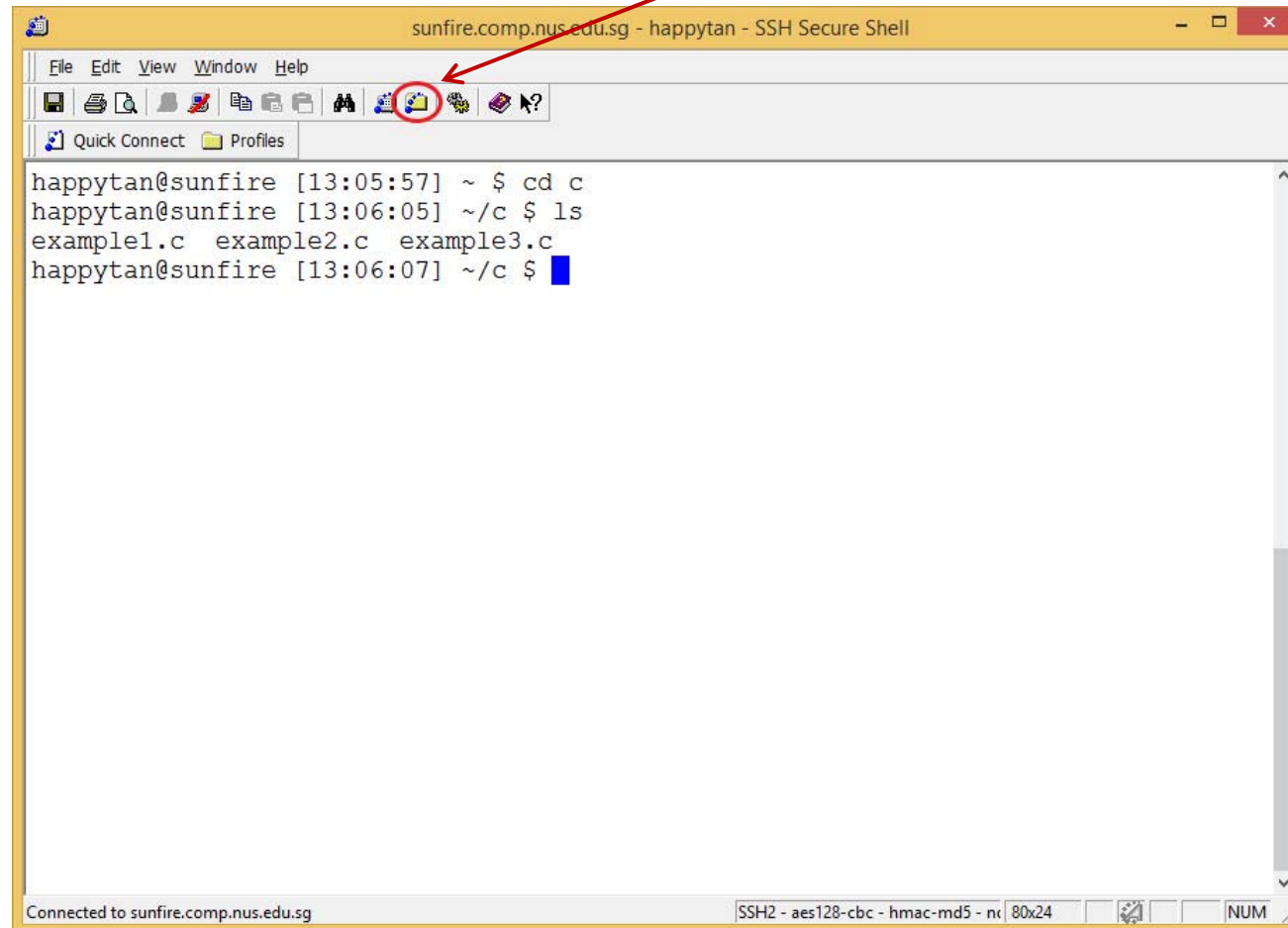
[illegible]

File Transfer (0/2)



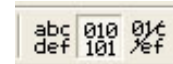
File Transfer (1/2)

- To transfer files between your sunfire account and your local computer, click on the **SSH Secure File Transfer** icon

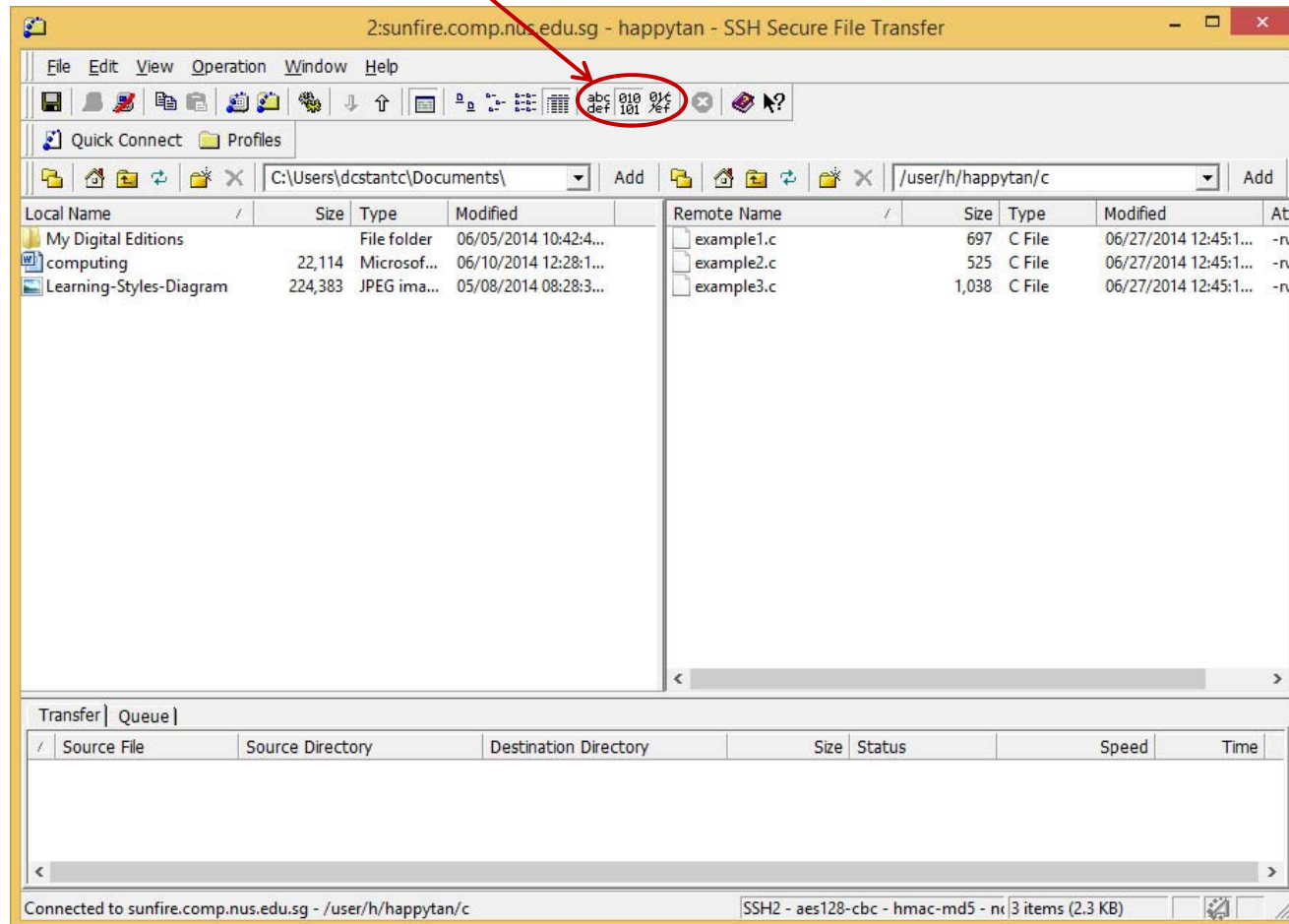


File Transfer (2/2)

- Left: your local machine; right: sunfire
- Choose the format: **ASCII**, **Binary** or **Auto**
- Click on file(s) to transfer, and drag to the destination

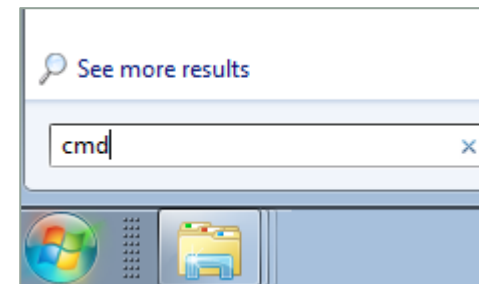


abc 010 01c
def 101 %ef



Using Samba Service (1/3)

- Samba service allows you to access your sunfire account directory on Windows **when you are connected to the SoC network (directly or via VPN).**
- Enable this service at the following URL
<https://mysoc.nus.edu.sg/~myacct/services.cgi>
- Open a command prompt on Windows
 - Click on the Start button
 - Type “cmd” in the search box, and press Enter



Using Samba Service (2/3)

- Type in the following command

`net use <drive> \\stusambahost\<UNIX id>`

- Replace <drive> with a drive letter of your choice
- Replace <UNIX id> with your UNIX id
- Example: `net use Z: \\stusambahost\happytan`
- Login with your NUSNET account when prompted
 - Username: nusstu\<NUSNET id>
 - Password: <NUSNET password>
- You will be able to access your sunfire account directory through the specified drive letter in Windows

Using Samba Service (3/3)

- If you are using your own PC,
 - Turn on SoCVPN @ <https://webvpn.comp.nus.edu.sg> using a non-Chrome browser (e.g., Firefox)
 - Use **stusambahost.comp.nus.edu.sg** in the **net use** command (instead of just stusambahost)

- For a detailed user guide on Samba service, visit <https://docs.comp.nus.edu.sg/node/1663>

Introductory Workshop

- After today's sectional class, please go through the document (http://www.comp.nus.edu.sg/~cs1010/labs/2016s1/intro_lab/gettingStarted.html) again and try out the commands yourself.
- If you think you still need guidance, please attend the Introductory Workshop. Details on registration will be posted on the **IVLE forum**.
- Objective:
 - To ensure that ALL students are ready to use the sunfire system, know basic UNIX commands, and able to edit, compile and execute C programs by next sectional class.

Summary

- In this unit, you have
 - Familiarised yourself with the **programming environment**
 - Accessed the sunfire system and learned some basic **UNIX commands**
 - Used the editor **vim** to create/modify your C programs
 - Used the compiler **gcc** to compile your C programs
 - Familiarised yourself with the **edit – compile – execute** process

End of File

ACKNOWLEDGEMENT

This presentation benefitted from

PowerPointLabs

a PowerPoint plugin for creating
better presentations with less effort.

PowerPointLabs
is **available for free** at
<http://PowerPointLabs.info>